



## EtherCAT als Antriebsbus

# Synchronität ist nicht genug

Wer hochdynamische Antriebstechnik über Ethernet realisieren will, darf nicht nur auf Jitter und Synchronität als Qualitätsparameter des Feldbusses achten; die Gleichzeitigkeit ist ebenso wichtig wie die Implementierung der etablierten Geräteprofile.



► Die speziellen Anforderungen der Antriebstechnik sind für ein Feldbussystem immer noch besonders schwierig zu erfüllen: Kurze Zykluszeiten und hohe Synchronität, wie sie über den Bus geschlossene Regelkreise benötigen, lassen sich in bisherigen Systemen nur mit speziellen Antriebsbussen realisieren, z. B.: Sercos-Interface (IEC 61491). Antriebsbusse eignen sich aber nur bedingt für vermeintlich triviale Aufgaben, wie I/O-Signale im Zyklus der SPS auszutauschen. Die Folge sind getrennte Feldbusse für Antriebe und I/O-Signale. Sind Steuerungen dann noch untereinander zu vernetzen, kommt Ethernet hinzu. Für den Anwender ergeben sich drei inkompatible Netzwerke mit jeweils unterschiedlicher Verkabelung und Parametrierung.

Aktuell aufkommende Ethernet-basierte Feldbusse versprechen hier Besserung. Alle Varianten nehmen für sich in Anspruch, antriebstechnikauglich zu sein, SPS-Signale verarbeiten zu können und natürlich die globale Vernetzung zu unterstützen – schließlich basieren sie auf Ethernet. Die Zusammenlegung allein reicht aber noch nicht; ein neues Feldbussystem sollte auch in den Details der un-

terschiedlichen Anforderungen deutliche Vorteile bringen.

### Definitions-Lücke geschlossen

Die vermeintlich harten Anforderungen der Antriebstechnik an ein Feldbussystem sind: Zykluszeit, Synchronität und Gleichzeitigkeit. Typische Werte für benötigte Zykluszeiten liegen zwischen moderaten 4 ms (zyklische Lagevorgabe mit Lageregelung im Antrieb) und extremen 62,5 µs (über den Bus geschlossener Stromregelkreis). Als ausreichende Synchronität wird häufig eine Mikrosekunde angegeben. Aber die Synchronität ist in diesem Zusammenhang nicht klar definiert. Daher sollte zusätzlich die Gleichzeitigkeit bewertet werden. Während die Synchronität den zeitlichen Jitter der Abarbeitung der Funktionen in den beteiligten Teilnehmern (Antriebe und Steuerung) angibt, definiert die Gleichzeitigkeit das Maß des zeitlichen Versatzes dieser Funktionen.

Synchronität ist für den einzelnen Teilnehmer wichtig, damit eigene, unterlagerte Regelkreise sich auf das zyklische Signal entsprechend genau synchronisieren können. Die Gleichzeitigkeit erlaubt

zudem, verteilte Teilnehmer an einer gemeinsamen Aufgabe, mit der absolut selben Zeitbasis, arbeiten zu lassen.

### Messtechnik setzt die Maßstäbe von morgen

Im Vergleich zu klassischer SPS-Technik, mit Zykluszeiten von rund 20 ms mit freilaufender Prozessdatenkommunikation, sind die beschriebenen Anforderungen schon deutlich höher. Moderne Steuerungen erreichen inzwischen aber die geforderten Zykluszeiten und tauschen ihre I/O-Signale ebenfalls synchron aus. Da eine SPS neben der eigentlichen Steuerungstechnik immer mehr Funktionen aus den Bereichen Überwachung und Messtechnik übernimmt, sind hier künftig sogar noch höhere Anforderungen zu erwarten. Maschinen- und Werkzeug-sicherungssysteme, die heute mit Hardware realisiert werden, lassen sich so als Software in die SPS integrieren.

#### AUTOR



DR. DIRK JANSSEN,  
Leiter der Abteilung  
Grundlagensoftware bei  
Beckhoff in Verl.

## KOMPAKT

In der Antriebstechnik begrenzt die Trägheit der anzusteuern Mechanik die benötigte Zykluszeit und Synchronität. Die bestimmenden Faktoren für das Einsatzspektrum eines Feldbussystems werden daher künftig eher aus der Messtechnik kommen.

### Zyklische Synchronisationssignale

Synchronität und Gleichzeitigkeit zwischen verteilten Systemen lassen sich auf verschiedene Art erreichen – zumindest ansatzweise. Ein häufig angewendetes Verfahren beruht auf zyklisch gesendeten Synchronisationssignalen. Das Verfahren setzt einen verzögerungsfreien Empfang in den Teilnehmern voraus und dass das Bussystem immer frei ist, wenn das Synchronisationssignal gesendet werden soll. Auf Grund von Laufzeiten in den Teilnehmern und in den Infrastrukturkomponenten (Switches, Netzwerk-Controller etc.) ist die Gleichzeitigkeit aber nicht gegeben: Ein 100 m langes Kabel erzeugt bereits eine Verzögerung von ca. einer halben Mikrosekunde, ein Switch bis zu mehrere Mikrosekunden.

### Synchronisation und Gleichzeitigkeit über verteilte Uhren

EtherCAT nutzt einen anderen Ansatz, der auf so genannten verteilten Uhren analog zu IEEE 1588 basiert: Alle Teilnehmer haben eine eigenständige Uhr, auf deren Basis die lokalen Zyklen und Ereignisse ablaufen. Entscheidend ist, dass alle Uhren gleich schnell laufen und die glei-

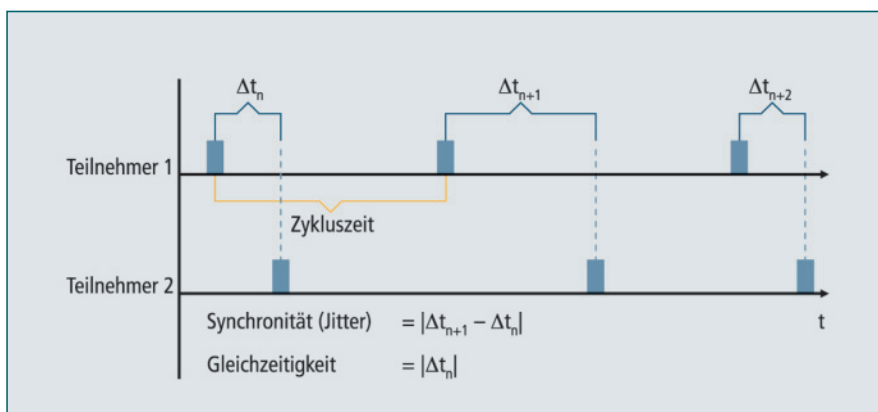
Kombinierte Einsatzfähigkeit für Antriebstechnik und schnelle I/Os – das war mit das wichtigste Entwicklungsziel bei EtherCAT. Dazu braucht es slaveseitig nicht nur ganz spezielle Eigenschaften, um Synchronität und Gleichzeitigkeit zu erzielen, sondern auch die Implementierung der relevanten Protokolle und Profile – neben CANopen kommt künftig auch Sercos zum Zug.

che Basiszeit haben. Dies stellt bei EtherCAT eine im Slave-Controller (ESC) integrierte Regelung sicher. Sie sorgt dafür, dass sich alle Uhren an einer Referenz-Uhr orientieren und unabhängig von Temperatur und Bauteil-Toleranzen synchron laufen.

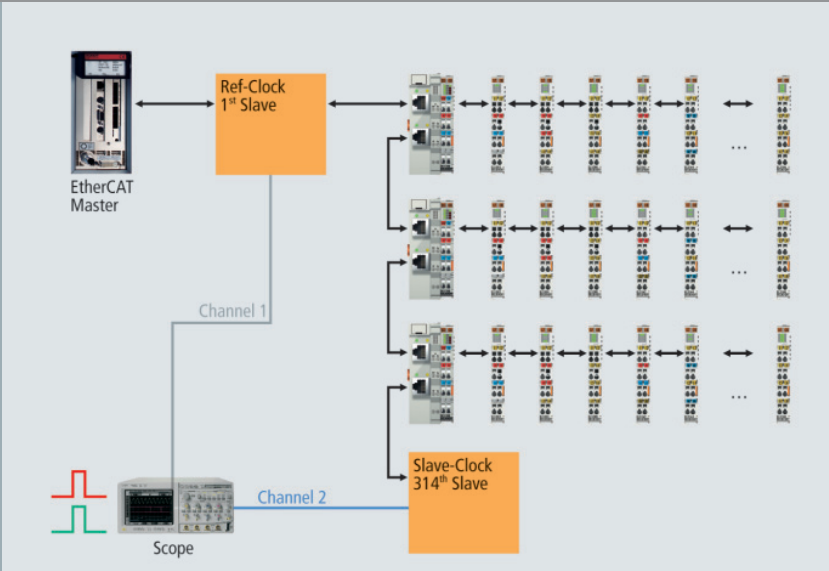
Um zusätzlich die Gleichzeitigkeit herzustellen, müssen alle Uhren die gleiche Basiszeit erhalten. EtherCAT definiert hierzu eine Systemzeit mit einer Auflösung von 1 ns. Die Systemzeit wird als 64-Bit-Integer direkt vom ESC verwaltet und stellt jederzeit den absoluten zeitlichen Bezug zu Ereignissen im gesamten Netzwerk her. Da durch die Regelung alle verteilten Uhren synchronisiert sind, muss die Systemzeit nur einmalig initialisiert werden.

### Signallaufzeiten berücksichtigt

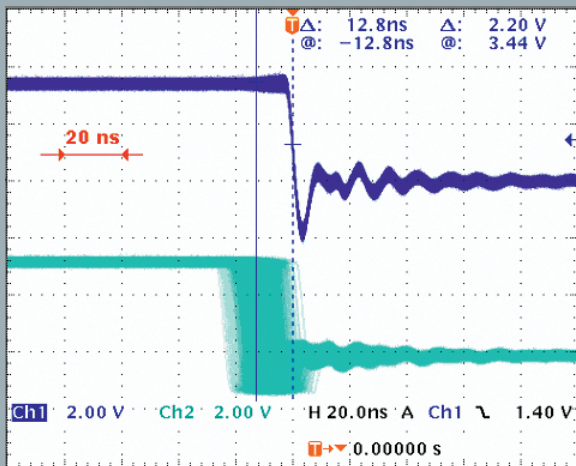
Um die Systemzeit entsprechend genau in allen Teilnehmern initialisieren zu können, wird eine vom ESC unterstützte



**Begriffsdefinition:** Neben kurzer Zykluszeit und hoher Synchronität (kleiner Jitter) muss ein Antriebsbus auch für Gleichzeitigkeit sorgen.



Realistisches Applikationsszenario: Testaufbau zur Ermittlung von Synchronität und Gleichzeitigkeit zweier EtherCAT-Teilnehmer, zwischen denen über 300 weitere Teilnehmer und insgesamt ca. 120 m Kabel liegen.



Testergebnisse: Das Scope triggert auf das Signal des ersten Teilnehmers und zeigt unten das Signal des zweiten. Die Breite der Signalflanke (Nachleuchtzeit auf unendlich) entspricht dem Synchronisationsfehler und liegt bei  $\pm 20$  ns. Die Abweichung des Mittelwertes vom Triggerpunkt des ersten Teilnehmers entspricht dem Gleichzeitigkeitsfehler und liegt hier bei ca. 12 ns.

Laufzeitmessung durchgeführt. Dazu ist u. a. die genaue Kenntnis der installierten Topologie notwendig, die bei EtherCAT sehr flexibel (Linie, Stern, Baum) aufgebaut sein kann. Die Topologie ist aber vollständig auslesbar. Eine statistische Mittelwertbildung hilft dabei, den im Kommunikationssystem vorhandenen Jitter zu kompensieren. Nach der Messung sind die Signallaufzeiten zwischen allen EtherCAT-Teilnehmern bekannt und werden bei der Verteilung der Systemzeit berücksichtigt.

Basierend auf den verteilten Uhren, bietet der ESC Funktionen und Signale, die die hochgenaue Zeitbasis für die jeweilige Anwendung (z. B. eines Antriebsreglers) zur Verfügung stellen. Gearbeitet wird dabei immer mit der absoluten Sys-

temzeit, die automatisch den zeitlichen Bezug verteilter Teilnehmer sicherstellt. Durch Synchronisationssignale, die z. B. mit Capture/Compare-Einheiten des Anwendungsprozessors verbunden sind, wird die hochgenaue Zeitbasis in der Anwendung verfügbar. Der ESC stellt auch Synchronisationseingänge zur Verfügung, mit deren Hilfe die Systemzeit für externe Ereignisse in den Speicher übernommen (gelatcht) wird. Durch Verwendung der absoluten Systemzeit können auch weit verteilte Ereignisse gemeinsam bewertet werden.

**Genauigkeit:  $\pm 20$  ns für Synchronität,  $\pm 12$  ns für Gleichzeitigkeit**  
Entscheidend für die Einsetzbarkeit der verteilten Uhren sind die Qualität der Re-

gelung und die Genauigkeit, mit der die Systemzeiten der einzelnen Teilnehmer initialisiert werden können. Die Synchronität von einer Mikrosekunde erreichen die klassischen, auf Antriebstechnik spezialisierten Feldbusse annähernd – die erreichte Gleichzeitigkeit liegt aber häufig deutlich darüber. Selbst Sercos-Interface vernachlässigt die erforderliche Gleichzeitigkeit und geht davon aus, dass alle Teilnehmer das entsprechende Synchronisationssignal zur selben Zeit empfangen. Real wird das Signal aber von einem Teilnehmer zum nächsten weitergeleitet und jeweils um ca.  $1 \mu\text{s}$  verzögert – bei nur 20 Teilnehmern sind das bereits  $20 \mu\text{s}$ . Dieser Fehler kann auch nicht automatisch kompensiert werden, da die Topologie bei Sercos nicht auslesbar ist, die Laufzeit nicht messbar ist und keine Vorkehrungen im entsprechenden Protokoll getroffen wurden.

EtherCAT ist dagegen ausgelegt, um nicht nur 20 Teilnehmer, sondern mehrere hundert bzw. tausend Teilnehmer zu unterstützen. Die speziellen Mechanismen sorgen trotzdem für Ergebnisse, die um mehrere Größenordnungen besser sind: Umfangreiche Messungen haben ergeben, dass sowohl der Synchronisationsfehler als auch der Gleichzeitigkeitsfehler – selbst bei großen Netzwerken – deutlich unter  $100$  ns liegen.

### Multiprotokollfähigkeit – Sercos und CANopen implementiert

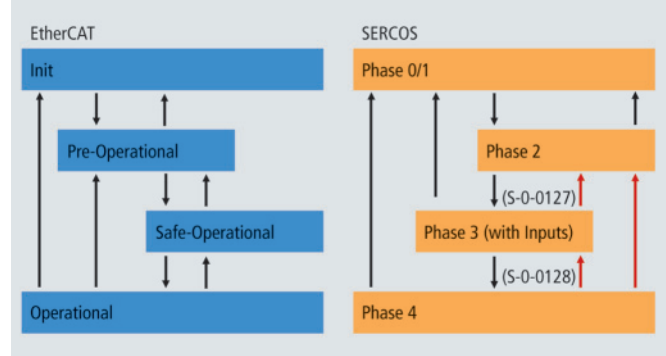
Weitere wichtige Kriterien eines Feldbus-systems zur Unterstützung der Antriebstechnik sind die verwendeten Geräteprofile. Diese sorgen für die Kompatibilität und den effizienten Datenaustausch zwischen Steuerung und Antrieb. Statt hier das Rad neu zu erfinden, setzt EtherCAT auf bewährte Technik.

Die Kommunikationsanforderungen moderner Feldbusse (Prozess-, Parameterdaten, paralleles TCP/IP, Firmware-Updates, Routing zu unterlagerten Bussystemen, etc.) werden von keinem verfügbaren Protokoll alleine unterstützt. Daher setzt EtherCAT auf Multiprotokollfähigkeit und führt die unterschiedlichen Pro-

tolle in einer einheitlichen Mailbox zusammen. Dies erleichtert u. a., bestehende Geräte schnell und vollständig auf EtherCAT umzusetzen. Für die Antriebstechnik relevant sind: CANopen over EtherCAT (CoE) und Sercos over EtherCAT (SoE). Protokolle wie Ethernet over EtherCAT (EoE) und File Access over EtherCAT (FoE) ermöglichen optional, z. B. einen Webserver im Antrieb zu integrieren oder die Firmware bzw. Kurvenscheibentabellen effizient über den Bus auszutauschen.

**Überschaubarer Migrationsaufwand**

Das CoE-Protokoll ermöglicht die Nutzung aller CANopen-Profiles – und damit auch des Antriebsprofils DS 402. Das SDO-Protokoll ist direkt übernommen, so dass bestehende CANopen-Stacks und darauf basierende Geräte quasi ohne Änderung in EtherCAT verwendet werden können. Optional sind Erweiterungen definiert, die einerseits die 8-Byte-Beschränkung aufheben und andererseits die vollständige Auslesbarkeit des Objektverzeichnisses ermöglichen. Die EtherCAT-Slave-State-Machine entspricht bis auf wenige Details der CANopen-State-Machine, so dass auch hier die notwendigen Änderungen überschaubar bleiben. Um ein eindeutigeres Anlaufverhalten zu ermöglichen, ist ein weiterer State namens 'Safe-Operational' definiert, in den bereits gültige Eingänge übertragen werden, während die



**EtherCAT-Slave-State-Machine und deren Abbildung auf die Sercos-Phasen**

Ausgänge noch im sicheren Zustand verbleiben.

Sercos over EtherCAT (SoE) erlaubt die Nutzung des bewährten, auf die anspruchsvolle Antriebstechnik spezialisierten Sercos-Profiles. Der Sercos-Servicekanal, und damit der Zugriff auf alle antriebsinternen Parameter und Funktionen, wird auf die EtherCAT-Mailbox abgebildet. Auch hier stehen sowohl die Kompatibilität zum bestehenden Protokoll als auch die Erweiterungsmöglichkeit bezüglich der Datenlängenbeschränkung im Vordergrund. Die Prozessdaten, bei Sercos-AT- und MDT-Daten, werden wiederum mit den Mitteln des EtherCAT-Slave-Controllers übertragen; das Mapping erfolgt Sercos-konform über die entsprechenden Identifier.

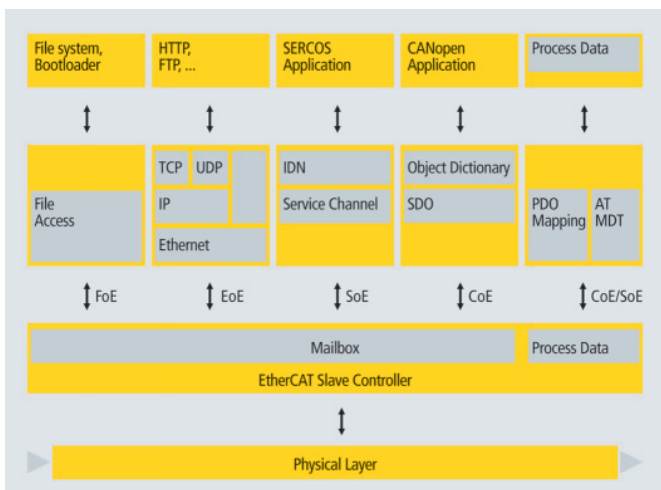
Die EtherCAT-Slave-State-Machine lässt sich ebenfalls auf die Phasen des Sercos-Protokolls abbilden. 'Pre-Operational' entspricht Sercos Phase 2 und erlaubt Servicekanalkommunikation ohne Prozessdatenaustausch. 'Safe-Operational' ist mit Phase 3 vergleichbar. Es findet die notwendige Synchronisierung statt, allerdings müssen bei EtherCAT bereits gültige Eingänge übertragen werden. 'Ope-

rational' entspricht wiederum exakt der Phase 4 (zyklischer Datenaustausch). Die EtherCAT-Slave-State-Machine ist auf einen Slave bezogen und erlaubt somit – im Gegensatz zu Sercos – einzelne Antriebe unabhängig von anderen Teilnehmern neu zu parametrisieren und in Betrieb zu nehmen.


**Drei Feldbusse auf einen Streich**

EtherCAT ist kein reiner Antriebsbus, erfüllt aber die entsprechenden Anforderungen mindestens um eine Größenordnung besser als darauf spezialisierte Systeme. Eine Trennung von Antriebs-, I/O- und Kommunikationsbus ist nicht mehr notwendig. Selbst anspruchsvolle Aufgaben aus der Messtechnik lassen sich integrieren und erlauben, neue Funktionen mit klassischer Steuerungstechnik zu kombinieren.

Durch Verwendung bewährter Kommunikationsprofile ist eine Migration bestehender Geräte und Applikationen einfach möglich. Außerdem bleiben so die gesamte Toolkette und vorhandene Erfahrungen zur Parametrierung entsprechender Antriebe erhalten. Erste realisierte Antriebe auf Basis der beschriebenen Protokolle werden bereits auf der SPS/IPC/Drives zu sehen sein.



**Multiprotokollfähigkeit der EtherCAT-Mailbox: Neben CANopen-Profilen wird auch das Sercos-Profil implementiert.**

<b>KONTAKT</b>	
<b>EtherCAT als Antriebsbus</b> <span style="float: right;"><b>779</b></span>	
Beckhoff <a href="http://www.beckhoff.com">www.beckhoff.com</a>	
 infoDIRECT	<b>779iee1104</b>
<ul style="list-style-type: none"> <li>▶ Servoverstärker für EtherCAT</li> <li>▶ Details zum Funktionsprinzip von EtherCAT</li> <li>▶ Der Ethernet-Feldbus</li> <li>▶ EtherCAT Technology Group</li> </ul>	